

Efficient ECC encryption for WSN's

Ravi Kishore Kodali¹, and Prof. Narasimha Sarma, N.V.S.¹

¹ National Institute of Technology, Warangal

Department of E. and C. E., Warangal, India

Email: ravikkodali@gmail.com

Abstract— Elliptic Curve Cryptography (ECC) provides a secure means of key exchange between communicating nodes using the Diffie-Hellman (DH) Key Exchange algorithm. This work presents an ECC encryption implementation using of the DH key exchange algorithm. Both encryption and decryption of text messages using this algorithm, have been attempted. In ECC, encoding is carried out by mapping a message character to an affine point on an elliptic curve. It can be observed from the comparison of the proposed algorithm and Koblitz's encoding method, that the proposed algorithm is as secure as Koblitz's encoding method and the proposed algorithm has less computational complexity as the encoding phase is eliminated altogether. Hence, energy efficiency of the crypto system is improved and the same can be used in resource constrained applications, such as Wireless sensor networks (WSNs). It is almost infeasible to attempt a brute force attack. The security strength of the algorithm is proportional to the key length. However, any increase in the key length results in more communication overhead due to encryption.

Index Terms— ECC, EC-DH, Koblitz encoding

I. INTRODUCTION

Wireless Sensor Networks (WSN's) have been finding their applications in various diversified fields ranging from commercial and industrial to military areas. In certain WSN applications, while the WSN data is transiting towards the base station (BS) using multi-hop connectivity comprising of wireless communication links among the nodes, the same data need to be sent in a secure manner. To meet this security requirement, the data need to be encrypted prior to its transmission by the sending node and the same cipher messages need to be decrypted upon reception by the receiving node. The same encryption and decryption can be achieved by adopting either symmetric key cryptographic (SKC) or public key cryptographic (PKC) algorithms. In SKC, the same key needs to be shared between the sender and the receiver beforehand. Each node, in the multi-hop communication path, needs to store the corresponding keys of all of its neighboring nodes. In order to provide better security, a PKC algorithm such as RSA algorithm is being widely used in most of the products and standards. However, in WSN applications, the RSA algorithm cannot be used due to the computational and energy constraints of the constituent nodes. Elliptic Curve Cryptography (ECC) is ideal for environments such as smart cards, WSN's [2],[3].

ECC offers performance advantage at higher security levels [6]. The principal advantage of the ECC compared to the RSA, is that it offers better security at reduced key sizes, as shown in Table I, thereby reducing processing overheads [7], [8]. ECC

makes use of elliptic curves (EC's), in which the variables and coefficients are all restricted to the elements of a finite field. Moreover, because of the apparent hardness of the underlying elliptic curve discrete logarithm problem (ECDLP) [4] and [5], ECC systems are also well suited for applications requiring security, which need to last longer. Each user taking part in public key cryptography uses a pair of keys: a public key and a private key [9]. Only that particular user knows the private key, whereas the public keys are distributed among all the users intending to communicate. Some public key algorithms may require a set of predefined constants to be known by all the devices taking part in the communication. In ECC, these predefined constants are also called domain parameters. An understanding of the ECC needs mathematical background on EC's [4].

TABLE I ECC AND RSA COMPARISON

ECC key length	RSA key length
160	1024
224	2048
256	3072
384	7680

$$y^2 = x^3 + ax + b, \quad (1)$$

where $4a^3 + 27b^2 \neq 0$.

Let $E(a, b)$ consisting of all the points (x, y) satisfying the equation (1) together with element at infinity O . A group can be defined based on the set $E(a, b)$ for specific values of a and b . The heart of ECC is discrete logarithm problem ECDLP that can be stated as it is computationally infeasible to find the value k such that $Q = kP$, where P and Q are known points on the elliptic curve. However, it should be relatively easy to find Q , where k and P are known. The first part of algorithm is to generate public and private keys by both the parties participating in the communication. Both the users should select a random base/ generator public point, G , on the elliptic curve, whose order is a prime, P . Each user generates a random secret integer less than the order of G . Public key of a particular user is the scalar multiplication of the user's secret integer and the generator point. Next phase is sharing a key using Diffie-Hellman algorithm, that provides secure key exchange. Finally, the message is hidden using this key and thus the message is encrypted. The rest of the paper is organized as follows: Section II presents algorithm description of the proposed algorithms, section III provides an implementation of the proposed algorithms, section IV gives a comparison of the proposed algorithm and the Koblitz's method and section V gives conclusions.

II. ALGORITHM DESCRIPTION

The equation defining an elliptic curve over a finite field, called Galois field, $GF(p)$ [4], is as follows:

$$y^2 \bmod p = (x^3 + ax + b) \bmod p, \quad (2)$$

where $4a^3 + 27b^2 \neq 0$, $(x, y) \in GF(p)$ and a, b are integers $< p$. An elliptic curve, E over $GF(p)$ consists of the solutions (x, y) defined by equation (2), along with an additional parameter called O , which is the point of infinity, a point on the EC. The basic elliptic curve operations are point addition and point doubling. ECC primitives require scalar point multiplication. Given a point, $P(x, y)$ on an EC, one needs to compute kP , where k is a positive integer. This is achieved by carrying out a series of doubling and addition operations over P .

Example:

If $k = 23$, then $kP = 23 \times P = 2(2(2(2P) + P) + P) + P$. The doubling and addition operations are carried out, depending on a sequence of operations determined for k . Public keys are generated using scalar multiplication of private keys with the generator point, which makes use of a series of doubling and addition operations.

Notation used

1. G is a Generator point with prime order m [10].
2. Key pair $\{P, n\}$, where P is a Public key $P = n * G$ and n is the Private key $< m$.
3. User Alice - U_{Alice} and User Bob - U_{Bob} .
4. \oplus is X - OR operation.
5. L - the no. of bits in the message to be sent at a time $L = \lceil \log_2(\text{message}) + 1 \rceil$, $V^1 = V \bmod 2^L$ (first L bits)

A. *Correctness*

$$P_{Bob} = n_{Bob} \times G \quad (3)$$

$$P_{Alice} = n_{Alice} \times G \quad (4)$$

$$n_{Alice} \times P_{Bob} = n_{Alice} \times n_{Bob} \times G = n_{Bob} \times P_{Alice} = S \quad (5)$$

$$\text{Cipher} = x^s \oplus \text{Message} \quad (6)$$

$$x^s \oplus \text{Cipher} = x^s \oplus x^s \oplus \text{Message} = \text{Message} \quad (7)$$

Encrypting a message in the point plays an important role in the security of the algorithm.

Algorithm 1 ALGORITHM WITH SIMPLE ENCRYPTION

1. Key Generation

A. Begin

B. Initiate the connection between users, U_{Alice} and U_{Bob}

C. $U_{Alice} = (P_{Alice}, n_{Alice})$ is the key pair for U_{Alice} .

$U_{Bob} = (P_{Bob}, n_{Bob})$ is the key pair for U_{Bob} .

D. U_{Bob} sends the point P_{Bob} to U_{Alice} . Similarly U_{Alice} sends P_{Alice} to U_{Bob} .

2. Encryption : Hiding

E. U_{Alice} computes the point $n_{Alice}(P_{Bob}) = S$

Let $S = (x_s, y_s)$.

U_{Alice} calculates Cipher $= x^s \oplus \text{Message}$.

U_{Alice} sends cipher to U_{Bob} .

3. Decryption

F. U_{Bob} computes the point $n_{Bob} \times P_{Alice}$, which is same as the point $S = (x_s, y_s)$. Finally U_{Bob} decrypts the message using

Message $= x^s \oplus \text{Cipher}$.

Algorithm 2 ALGORITHM WITH COMPLEX ENCRYPTION

1. Key Generation

A. Begin

B. Initiate the connection between users, U_{Alice} and U_{Bob}

C. $U_{Alice} = (P_{Alice}, n_{Alice})$ is the key pair for U_{Alice} .

$U_{Bob} = (P_{Bob}, n_{Bob})$ is the key pair for U_{Bob} .

D. U_{Bob} sends the point P_{Bob} to U_{Alice} . Similarly U_{Alice} sends P_{Alice} to U_{Bob} .

2. Encryption : Hiding

E. U_{Alice} computes the point, S_1 , and let $S_1 = (x_{s1}, y_{s1})$

$= n_{Alice1}(P_{Bob1}) = n_{Alice1}(P_{Bob2}) = S_2$ Let $S_2 = (x_{s2}, y_{s2})$

F. U_{Alice} calculates the distance between S_1 and S_2 . Let it be D .

G. U_{Alice} calculates $V_1 = x_{s1}$ and

$V_2 = x_{s2}$. Cipher $= V_1 \oplus V_2 \oplus \text{Message}$.

H. U_{Alice} sends cipher to U_{Bob} .

I. For the next session, Alice changes the values to

$V_1 = V_1 + D, V_2 = V_2 + D$.

3. Decryption

J. U_{Bob} computes the point and S_2 .

K. U_{Bob} calculates $V_1 = x_{s1}$ and $V_2 = x_{s2}$.

L. Finally, U_{Bob} decrypts the message using Message $= V_1 \oplus V_2 \oplus \text{Cipher}$. M

For the next session, Bob changes the values to

$V_1 = V_1 + D, V_2 = V_2 + D$.

III. IMPLEMENTATION OF THE PROPOSED ALGORITHM

After defining the EC parameters, we can select a base point G . G has (x, y) coordinates satisfying the equation (2). The base point has the smallest x, y values which satisfy the elliptic curve EC. The ECC method requires that we select a random integer k , which needs to be kept secret. Then, kG is evaluated, by a series of addition and doubling operations, as discussed. For the purpose of this discussion, consider the source as host Alice, and the destination as host Bob. The private key of the host Bob is selected and it is n_{Bob} . Bob's public key is computed using $P_{Bob} = n_{Bob} \times G$. Similarly, Alice also computes her public key, $P_{Alice} = n_{Alice} \times G$ and both of these public keys are exchanged and a secret point, S , is computed by both independently.

A. Simple Encryption

Suppose Alice wants to encrypt and transmit a character to Bob, she does the following: Assume that host Alice wants to transmit the character C . Then the ASCII value of the character 'C' is used to modify the secret point, thereby encrypting the message somehow into the secret point. The encryption process is the x-or operation between the first n bits of the x-coordinate of a secret point, S , and the message to be transmitted, where n is the number of bits in the message. Simple encryption is illustrated in Fig. 1

The Elliptic curve is

$$y^2 \bmod 1021 = (x^3 + 3x + 16) \bmod 1021 \quad (8)$$

The base point G is selected as $(4, 33)$. The base point implies that it has the largest order and smaller x, y co-ordinates satisfying the EC equation. The order of G is $m = 1058$. $n_A = 83$

and $n_B = 127$. $P_B = 127(4, 33) = (900, 460)$. The secret point is $S = n_A \cdot P_B = 83(900, 460)$. $L = 8$ -bits in message. Therefore, $x^s = 190 = 10111110$

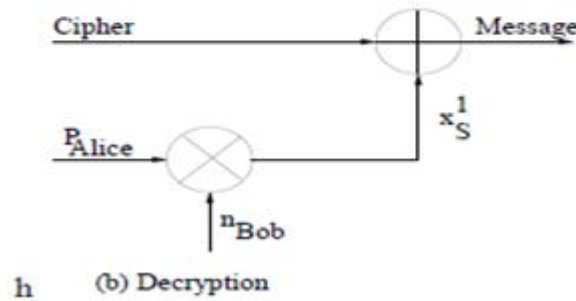
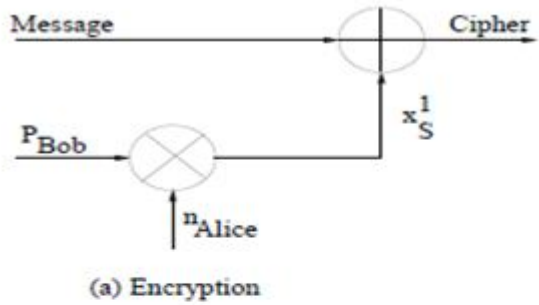


Figure 1 : Simple encryption

SIMPLE ENCRYPTION

Encryption:

Plaintext character, 'C', its ASCII value is 67.

Therefore, message = 67 = 01000011.

Cipher = $x^s \oplus \text{Message} = 01001000 = 72 = H$

Decryption:

Cipher = H = 01001000. Message = $x^s \oplus \text{Cipher} = 01000011 = C$.

Table II provides the cipher values for the sample message, "DECIPHER" using simple encryption technique.

TABLE II. SIMPLE ENCRYPTION

Message	ASCII	Cipher
D	68	250
E	69	251
C	67	253
I	73	247
P	80	238
H	72	246
E	69	251
R	82	236

B. Complex Encryption

The elliptic Curve is

$$y^2 \bmod 487 = (x^3 - 5x + 25) \bmod 487 \quad (9)$$

$G = (2, 5)$. Suppose $n_{Alice1} = 63, n_{Alice2} = 93, n_{Bob1} = 71, n_{Bob2} = 53$.

$P_{Alice1} = 63 \times G = (139, 347)$,

$P_{Alice2} = 93 \times G = (486, 121)$,

$P_{Bob1} = 71 \times G = (302, 57)$ and

$P_{Bob2} = 53 \times G = (176, 76)$.

According to the algorithm, $S_1 = (x_{s1}, y_{s1}) = (11, 422)$ and $S_2 = (x_{s2}, y_{s2}) = (152, 126)$

The distance between these points, S_1 and S_2 is computed

$$D = ((152 - 11)^2 + (422 - 126)^2)^{1/2} = 327 = 101000111$$

$$D^1 = 71 = 01000111$$

$$V_1 = x_{s1} = 11 = 00001011$$

$$V_2 = x_{s2} = 152 = 10011000$$

Fig. 2 illustrates Complex encryption method.

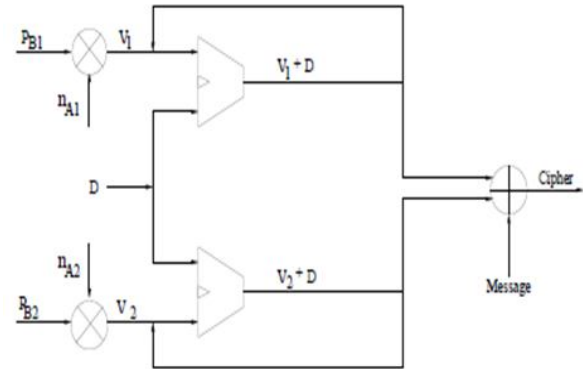


Figure 2 : Complex Encryption

Encryption:

Plaintext character is 'C', its ASCII value is 67.

Therefore, Message = 67 = 01000011

Cipher = $V_{11} \oplus V_{21} \oplus \text{Message} = 11010000 = 208$ Decryption:

Cipher = 11010000

Message = $V_{11} \oplus V_{21} \oplus \text{Cipher} = 01000011 = C$

For the next session,

$$V_1 = V_1 + D = 11 + 71 = 82 = 01010010$$

$$V_2 = V_2 + D = 152 + 71 = 223 = 11011111.$$

The elliptic curve as shown in Fig. 3 is

$$y^2 \bmod 487 = (x^3 - 5x + 25) \bmod 487 \quad (10)$$

$G = (0, 5)$ order, $m = 825$.

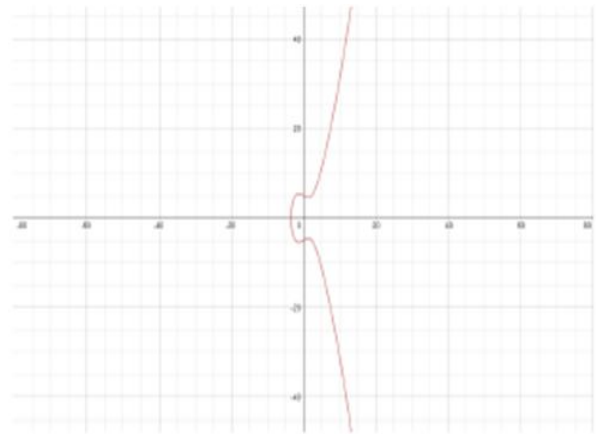
Figure 3: Elliptic curve, E: $y^2 = x^3 - 5x + 25$

TABLE III. Complex Hiding

Message	V_1	V_2	Cipher
D(68)	25	9	84
E(69)	12	252	181
C(67)	255	239	83
I(73)	242	226	89
P(80)	229	213	96
H(72)	216	200	88
E(69)	203	187	53
R(82)	190	174	66

Suppose, $nA1 = 41$, $nA2 = 79$, $nB1 = 32$, $nB2 = 68$.

$$PA1 = 41 \times G = (345, 334),$$

$$PA2 = 79 \times G = (104, 183),$$

$$PB1 = 32 \times G = (25, 261)$$

$$PB2 = 68 \times G = (295, 340).$$

Shared keys, $S_1 = (25, 226)$ and $S_2 = (265, 264)$.

From the shared keys, S_1 and S_2 ,

$$V_1 = 00011001 = 25, V_2 = 00001001 = 9,$$

$$\text{The distance, } D = ((25 - 265)^2 + (226 - 264)^2)^{1/2} = 243.$$

A sample sequence of characters "DECIPHER" is sent. The complex hiding method is illustrated in Table-III with an example using a series of characters "DECIPHER". The encryption of each character involves two 8-bit addition and two 8-bit X-OR operations. The time elapsed for encryption of single character or 8-bit data, implemented in MATLAB, is 0.000008 seconds. Therefore, the CPU time elapsed for encryption of the message DECIPHER is $8 \times (0.000008) = 0.000064$ seconds.

The complex encryption method typically consumes more power, when compared with simple encryption method, but less than any of the other ECC encryption techniques, which make use of an additional encoding process. An example of such an algorithm is ECC encryption using Koblitz's method of encoding [1]. Complex encryption of a message provides the algorithm with efficient security. It is almost infeasible to attempt brute force attack. Since encryption involves XOR operations consuming less computational time, power with efficient security.

IV. COMPARISON

A. Time and Power consumption

ECC encryption with Koblitz's method of encoding is one of the best encryption algorithms that provides reliable security. Such algorithms consume additional power for encoding and encryption, whereas our process does not consume so much power. The plot given in Fig. 4 shows the time consumed for encryption for various lengths of primes.

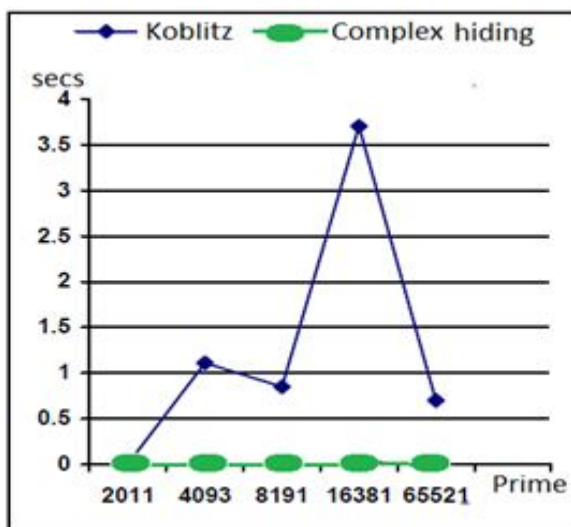


Figure 4: Comparison between Koblitz encoding [1] and Complex Encryption algorithm

B. Security

Though simple encryption is less secure, when compared with the complex encryption, but for appropriate prime lengths, the breaking process by a brute force attack can consume hundreds of years for completion. It is almost infeasible to attempt brute force attack in the case of complex encryption process. In order to break this algorithm, an intruder requires the knowledge of the three parameters: V_1 , V_2 and D .

Assumption of either of the V_1 and V_2 values requires breaking of ECDLP, which is almost infeasible for higher key lengths. Since D is a step function, tracing points back from D (if known), is also a difficult task. Also, the change in values V_1 and V_2 after each session, adds more security for the message. It is impossible to realize the above parameters from any of the known ciphers.

V. SECURITY ANALYSIS

A. Advantages of complex encryption

- Two keys used for encryption results in increased sample space of cipher text.
- Shared secret key is calculated over elliptic curve and hence private key cannot be retrieved by attacker because of elliptic curve discrete logarithmic problem
- Computationally intensive elliptic curve arithmetic operations are carried out only for key calculation. Message encoding to elliptic curve is not necessary as in case of Koblitz's method.
- Iterative key update assures forward secrecy.
- Simple Ex-OR operation used for encryption and decryption reduces complexity.
- Distance calculated between two secret keys is non-linear with keys used for encryption and decryption and hence improves key update mechanism.

B. Limitations

- Message confidentiality is reduced compared to Koblitz's encoding because encryption does not impose elliptic curve discrete logarithm problem for each encryption.
- Only part of shared secret key calculated over elliptic curve is used for encryption and decryption. Hence, security provided by ECDH is not completely utilized.

V. CONCLUSIONS

A plaintext message DECIPHER is used for implementing the algorithm proposed in this paper. Each character in the message is represented by its ASCII value and then encryption is carried out using both the algorithms. The execution time for encoding and decoding functions is no more required. The execution time taken for encryption is constant for different values of a , b , P for particular key length. Range of message bits that can be sent per single cipher is equal to the key length. As the key length increases, the security also increases exponentially due to the complexity in ECDLP. Since this algorithm consumes less power for encryption, it can be used in resource constrained applications, such as, WSN's. The energy consumption due to the security overhead can be

reduced, thereby extending the life of each sensor node.

REFERENCES

- [1] B. Padma, D. Chandravathi, and P. Roja, "Encoding and decoding of a message in the implementation of elliptic curve cryptography using koblitz method," *International Journal on Computer Science and Engineering*, vol. 2, pp. 1904–1907, 2010.
- [2] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *Wireless Communications, IEEE*, vol. 11, no. 1, pp. 62–67, 2004.
- [3] X. Tian, D. Wong, and R. Zhu, "Analysis and improvement of an authenticated key exchange protocol for sensor networks," *Communications Letters, IEEE*, vol. 9, no. 11, pp. 970–972, 2005.
- [4] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*. Springer, 2004.
- [5] W. Stallings, *Network and internetwork security: principles and practice*. Prentice-Hall, Inc., 2003.
- [6] M. Hasan, "Power analysis attacks and algorithmic approaches to their countermeasures for koblitz curve cryptosystems," *Computers, IEEE Transactions on*, vol. 50, no. 10, pp. 1071–1083, 2001.
- [7] Q. Qiu and Q. Xiong, "Research on elliptic curve cryptography," in *Computer Supported Cooperative Work in Design*, 2004. Proceedings. The 8th International Conference on, vol. 2, may 2004, pp. 698 – 701 Vol.2.
- [8] M. Khabbazi, T. Gulliver, and V. Bhargava, "Double point compression with applications to speeding up random point multiplication," *Computers, IEEE Transactions on*, vol. 56, no. 3, pp. 305–313, 2007.
- [9] P. Longa and A. Miri, "Fast and flexible elliptic curve point arithmetic over prime fields," *Computers, IEEE Transactions on*, vol. 57, no. 3, pp. 289–302, 2008.
- [10] R. Ramasamy, M. Prabakar, M. Devi, and M. Suguna, "Knapsack based ecc encryption and decryption," *International Journal of Network Security*, vol. 9, no. 3, pp. 218–226, 2009.